GAUSSIAN HitFinder

LArSoft Meeting

Jonathan Asaadi

Syracuse University

4/25/12

Outline

- Motivation of Looking into HitFinder
  - Existing issues with FFTHitFinder
  - Thing that have changed since talk on 4/18/12 ***(See reconstruction meeting talk)***
- Propose a “new” HitFinder Algorithm ***(RECAP)*** (GausHitFinder & GausHitFinderAna)
- Preliminary look at GausHitFinder Performance

  (Bug fixes from last time)
  - Side-by-Side comparison with FFTHitFinder
- Next-steps / Conclusions

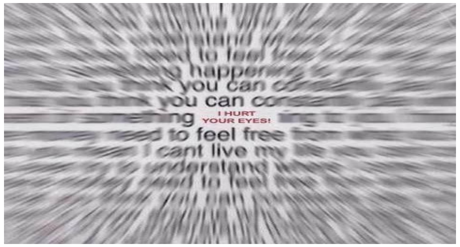
Motivation for Looking into HitFinder

(Reminder)

- **There has been a lot of effort going into our reconstruction code**
  - Some of the problems faced by the reconstruction algorithms could have their root in poor quality hits

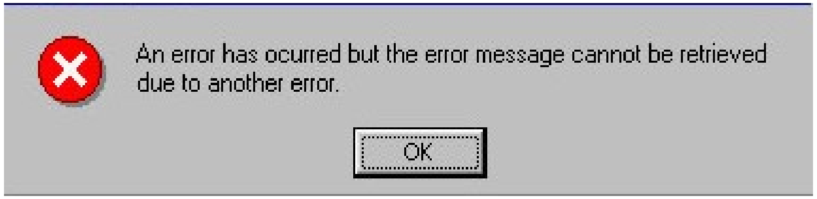
“Problems” with FFTHitFinder

- **Code was difficult to read**



- **Errors reported back on the hits didn't make sense**

(see back-up slide)



An error has ocurred but the error message cannot be retrieved due to another error.

  - **The multiplicity of the hit (# of peaks) wasn't being reported**

What has changed since last time

- **Minor modification to the code to use the same TF1 functions as FFTHitFinder**
  - This allows me to better handle multi-peaked hits that are merged together

- **Minor changes in the method I was using to “seed” the position of the hit**
  - Closer match what the original FFTHitFinder was doing, still maintains the error reporting of the fit and determining the multiplicity of the hit

- **Use the multi-Gaussian functions to calculate the “charge” using the 'area' methods**
  - Matches the functionality of the FFTHitFinder

“New” Hit Finding Algorithm
(GausHitFinder)

Keep all the good parts of the FFTHitFinder!

- Finding local minima and maxima
- Keep the same interface and data members with the Hit Reco object

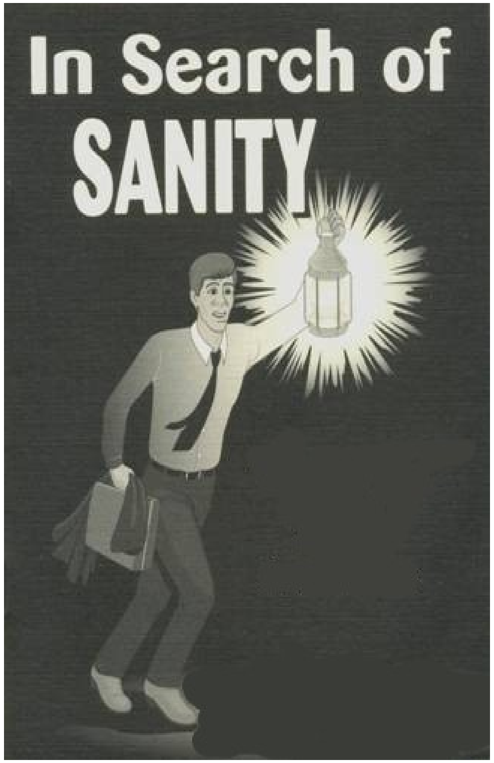
Make improvements to the algorithm

- Change the fitting procedure slightly *(see back-up slides)*
- Rewrite the code to be more “user friendly”
- Allow rejection of “bad” hits based on χ^2 / NDF

Results of the GausHitFinder Algorithm

Single Particle μ [2.0 GeV]

Event #	# of Hits Found (GausHitFinder)	# of Hits Found (FFTHitFinder)
1	5521	5528
2	5812	5824
3	5835	5838
4	5552	5556
5	5499	5550
6	5762	5771
7	5713	5717
8	5740	5746
9	5668	5672
10	5794	5805
Totals	**56896**	**57007**



GausHitFinder finds 99.8 % of the same hits as FFTHitFinder

⟶ *This 0.2 % difference comes entirely in the multi-peaked pulses*

Results of the GausHitFinder Algorithm

Genie Events

Event #	# of Hits Found (GausHitFinder)	# of Hits Found (FFTHitFinder)
1	3864	3869
2	14811	14904
3	3829	3845
4	0	0
5	8287	8295
6	2761	2767
7	2771	2777
8	0	0
9	8582	8640
10	4462	4479
Totals	**49376**	**49576**

GausHitFinder finds 99.6% of the same hits as FFTHitFinder

⟶ *This 0.4 % difference comes from multi-peaked hits*

Results of the GausHitFinder Algorithm

Single Particle μ [2.0 GeV]

What is the time performance of the two algorithms?

FFTHitFinder

(Running over 10 single muon events)

Avg. Time
~ 14 seconds per event

GausHitFinder

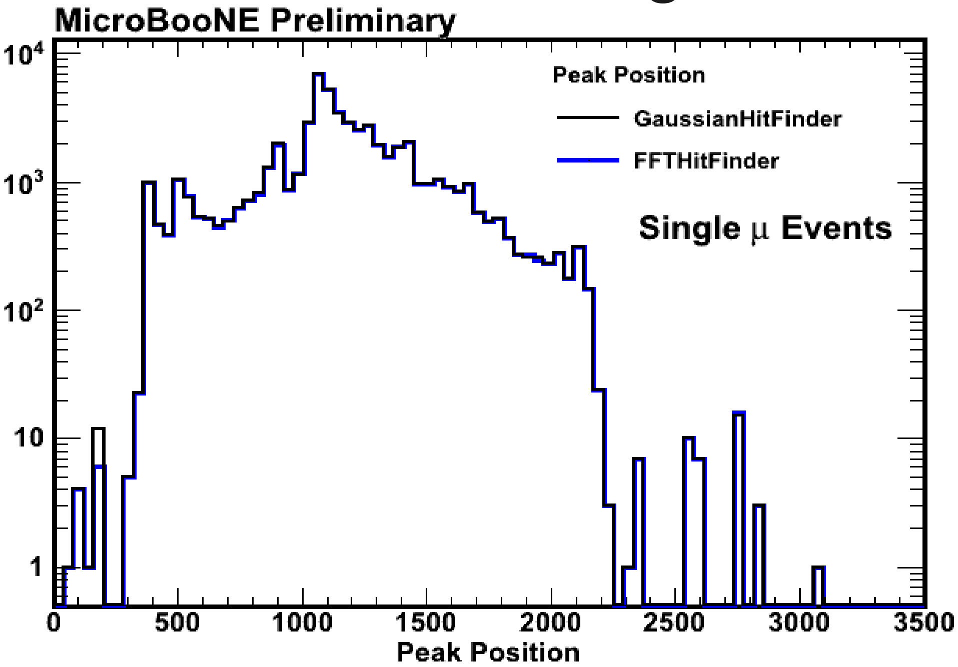
(Running over 10 single muon events)

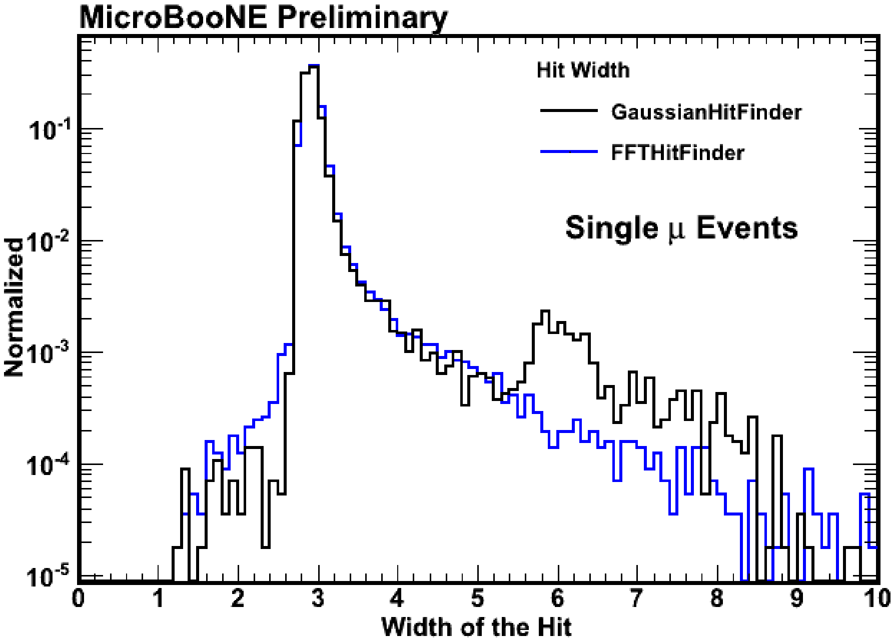
Avg. Time
~ 40 seconds per event

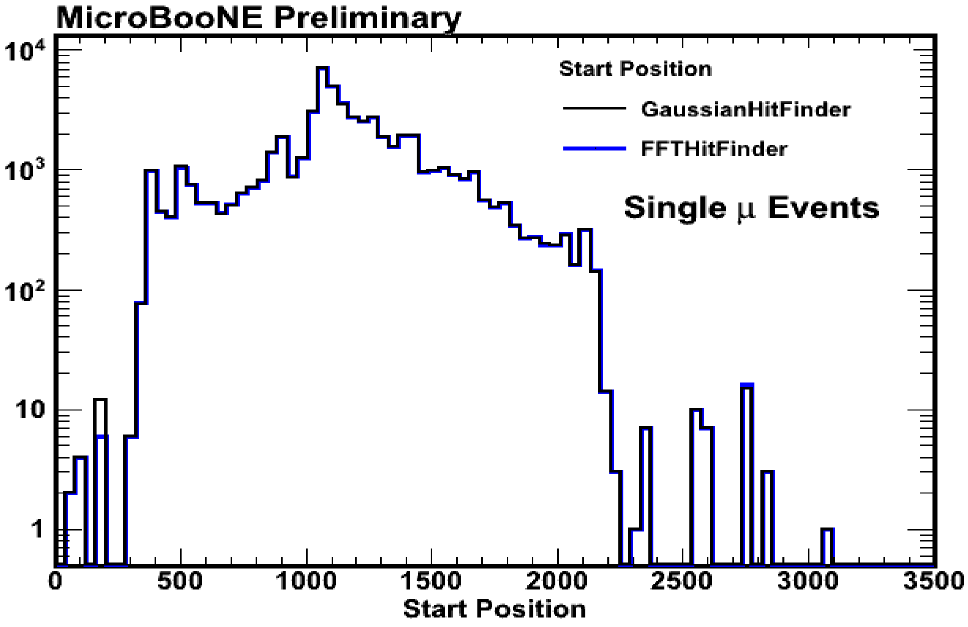
A difference in performance time...but could still be improved (work in progress)

Results of the GausHitFinder Algorithm

Single Particle μ [2.0 GeV]



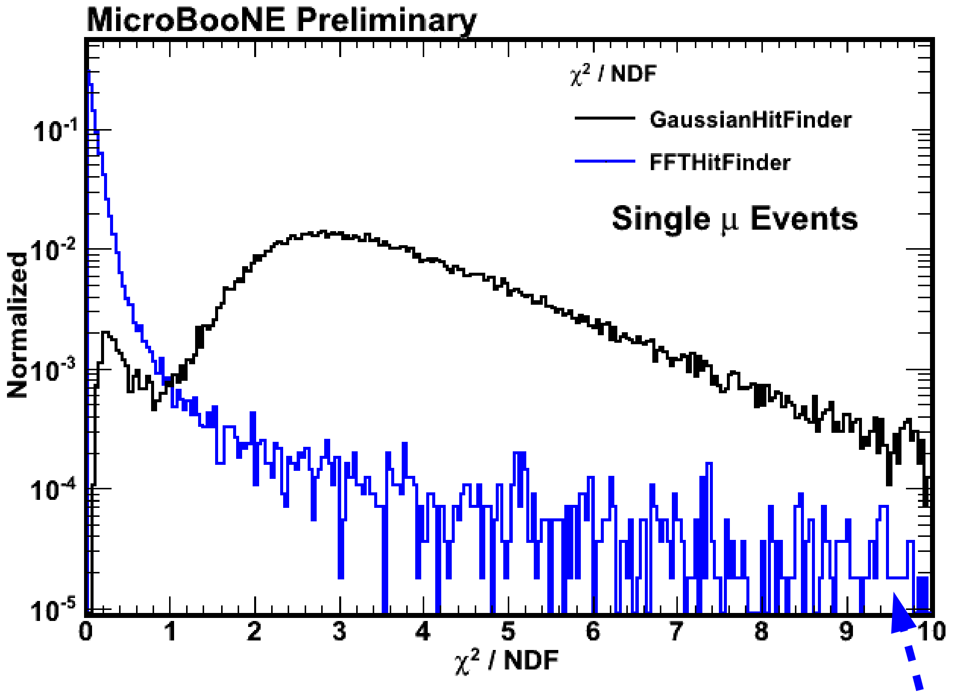




Very good agreement between the two algorithms

Results of the GausHitFinder Algorithm

Single Particle μ [2.0 GeV]



Recall:

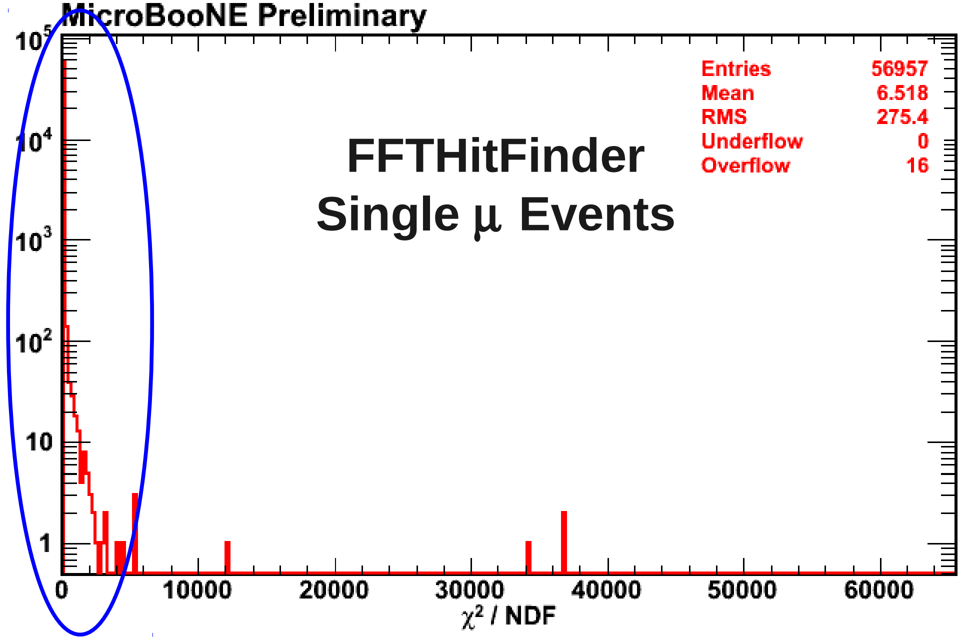
The χ^2 / NDF for the FFTHitFinder seemed to be broken

χ^2 / NDF for the FFTHitFinder range from 60,000 to much less then one!

χ^2 / NDF values make much more sense for the fits performed

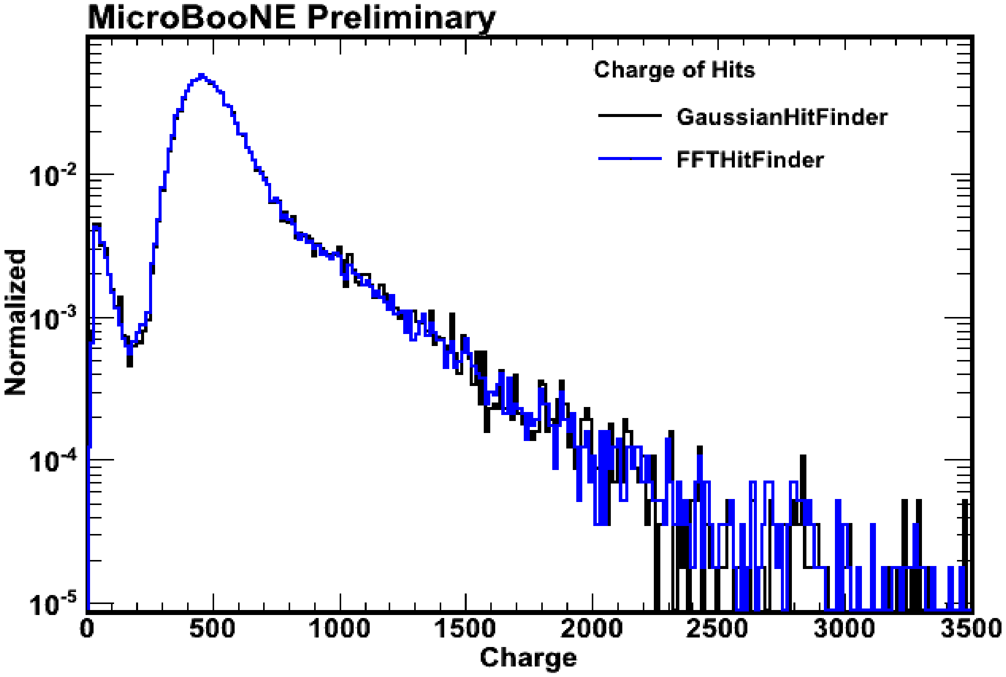
Possibly suggests having a loose cut to throw out large χ^2 / NDF

(Require χ^2 / NDF < 10)



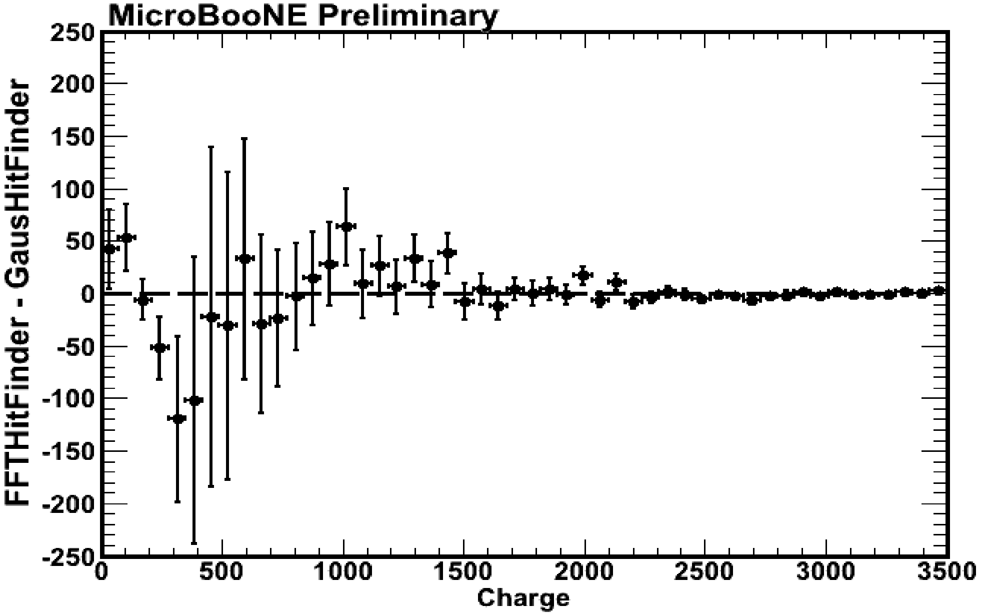
Results of the GausHitFinder Algorithm

Single Particle μ [2.0 GeV]



Looking at the charge found by the two algorithms

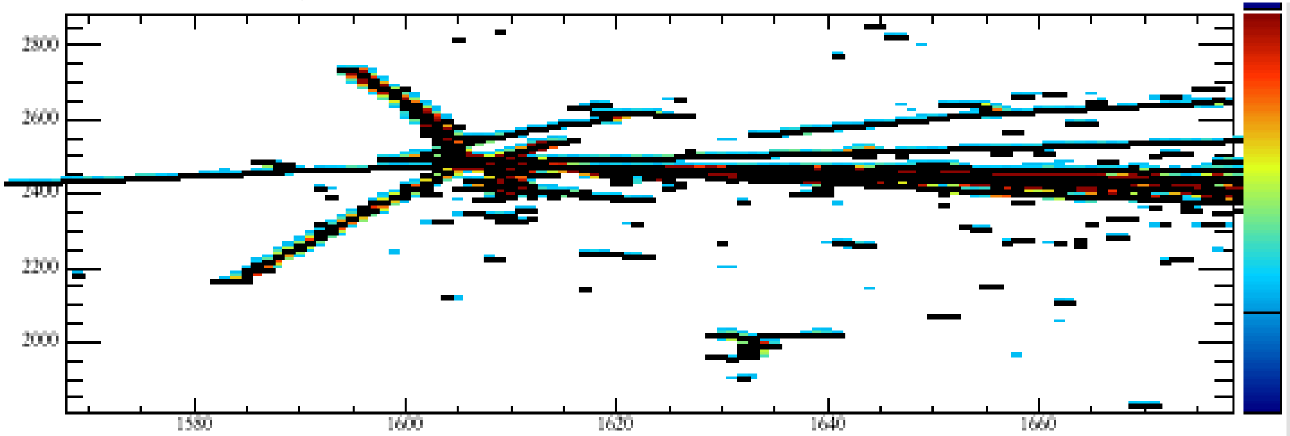
Matches the results of the FFTHitFinder fairly well

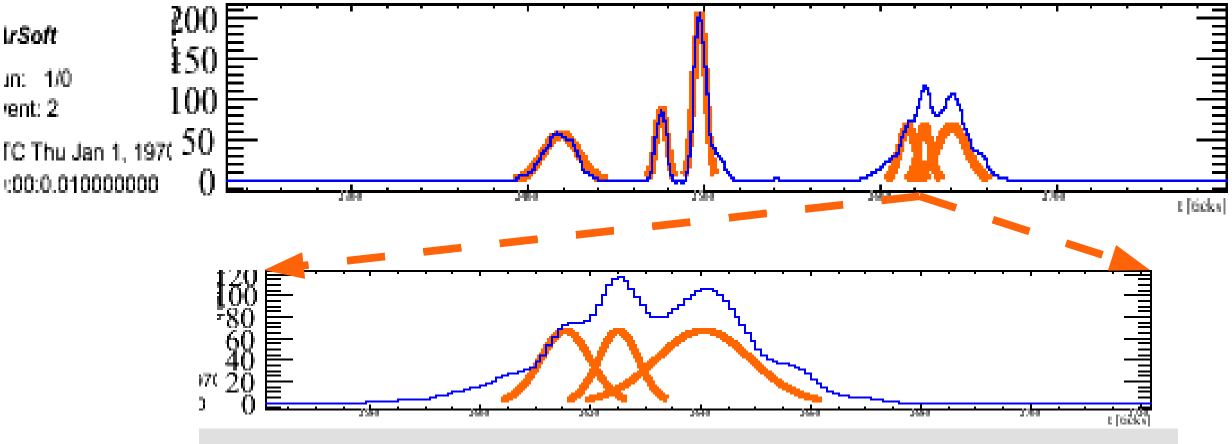


Results of the GausHitFinder Algorithm

Single Particle μ [2.0 GeV]

October - December 2012
multiplicity
the must-have magazine for all moms of multiples.
What to do when they ask why?
Yes, you can!
Capture moments without a meltdown
+
Hottest holiday toys for multiples
Survive & keep the miracle alive!
www.multiplicitymag.com



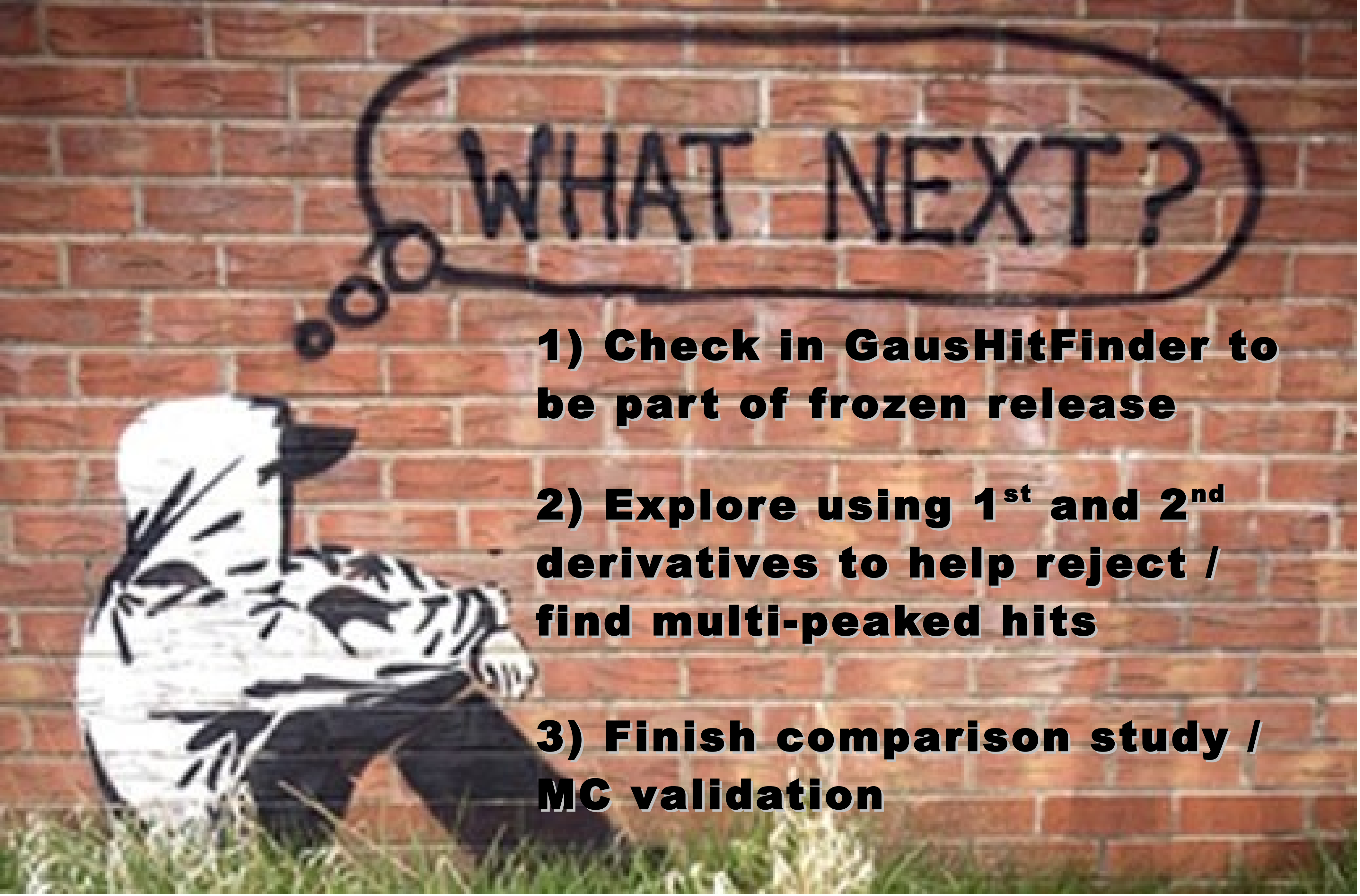


GausHitFinder now can handle multi-peaked hits as well as correctly identify the multiplicity of the hit

→ Can allow us to identify how to handle "Goodness of fit" for high multiplicity hits

→ Beginning to look at using the derivative of the pulse / hit to determine how we should fit

(Thanks to T. Junk for suggestion!)



1) Check in GausHitFinder to be part of frozen release

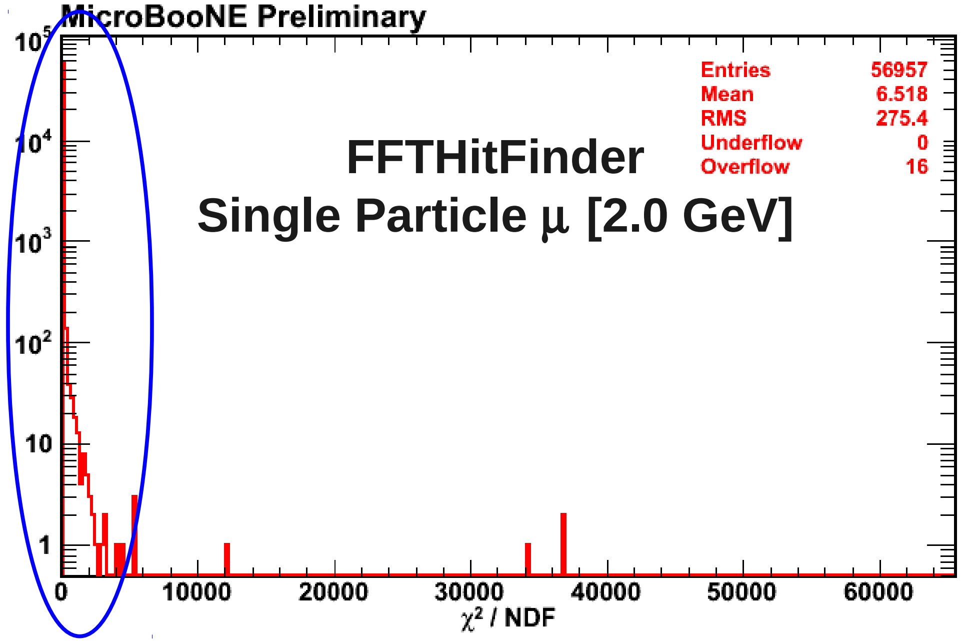
2) Explore using 1st and 2nd derivatives to help reject / find multi-peaked hits

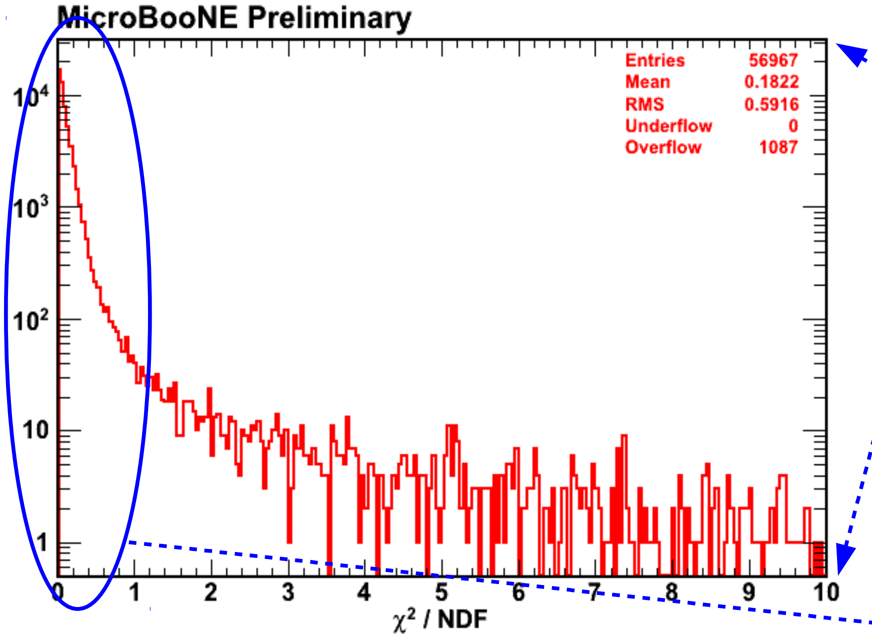
3) Finish comparison study / MC validation

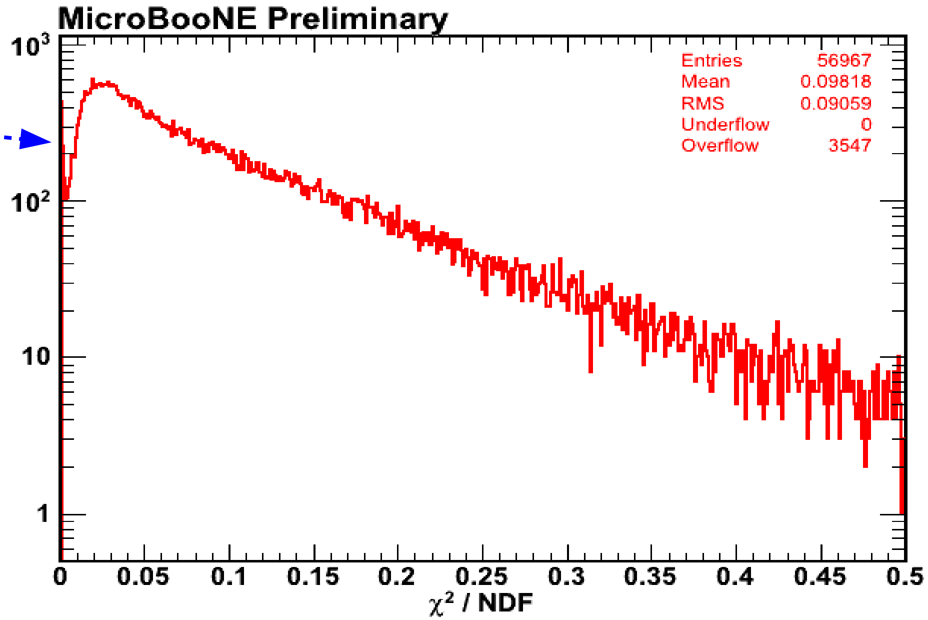
Back-up Slides

FFTHitFinder “Problems”

Looking at the χ^2 / NDF for the hits found the values range from 60,000 to much less then one!



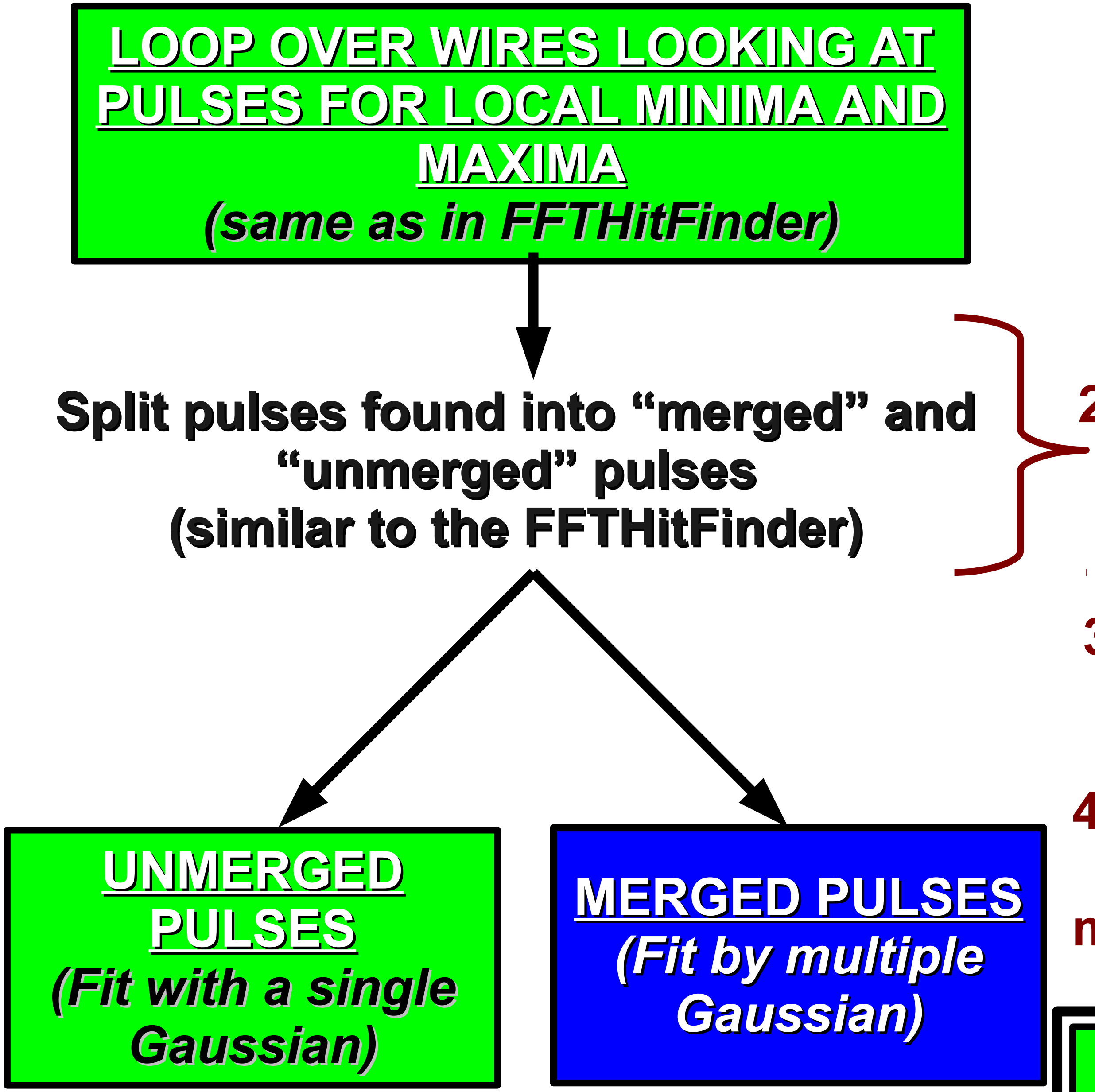




This seems to be an artifact of the way the fitting is done in the FFTHitFinder and makes evaluating these hits very difficult

***Note**: FFTHitFinder lacks the ability to seperate out single peak pulses and multi-peak pulses*

GausHitFinder Algorithm



LOOP OVER WIRES LOOKING AT PULSES FOR LOCAL MINIMA AND MAXIMA
(same as in FFTHitFinder)

Split pulses found into "merged" and "unmerged" pulses (similar to the FFTHitFinder)

UNMERGED PULSES
(Fit with a single Gaussian)

MERGED PULSES
(Fit by multiple Gaussian)

1) If there is no gap between the end of the previous pulse and the start of the next

2) If the height of the minimum is greater then ½ the ADC threshold for hits

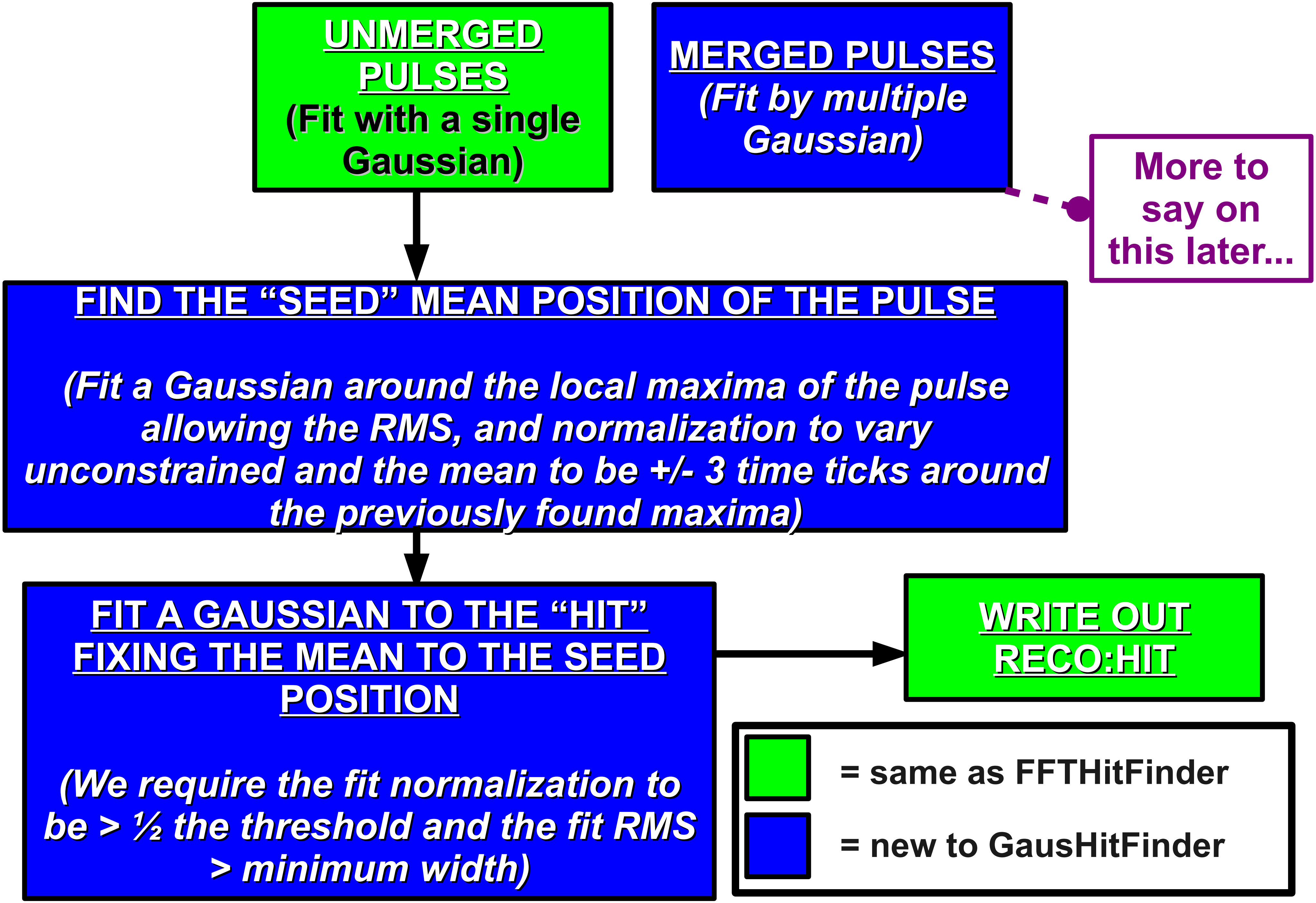
3) If the pulse is not at the end of the wire

4) If the number of consecutive pulses is less than the # of maximum consecutive hits (i.e. = 3 by default)

🟩 = same as FFTHitFinder

🟦 = new to GausHitFinder

GausHitFinder Algorithm



UNMERGED PULSES
(Fit with a single Gaussian)

MERGED PULSES
(Fit by multiple Gaussian)

More to say on this later...

FIND THE "SEED" MEAN POSITION OF THE PULSE

(Fit a Gaussian around the local maxima of the pulse allowing the RMS, and normalization to vary unconstrained and the mean to be +/- 3 time ticks around the previously found maxima)

FIT A GAUSSIAN TO THE "HIT" FIXING THE MEAN TO THE SEED POSITION

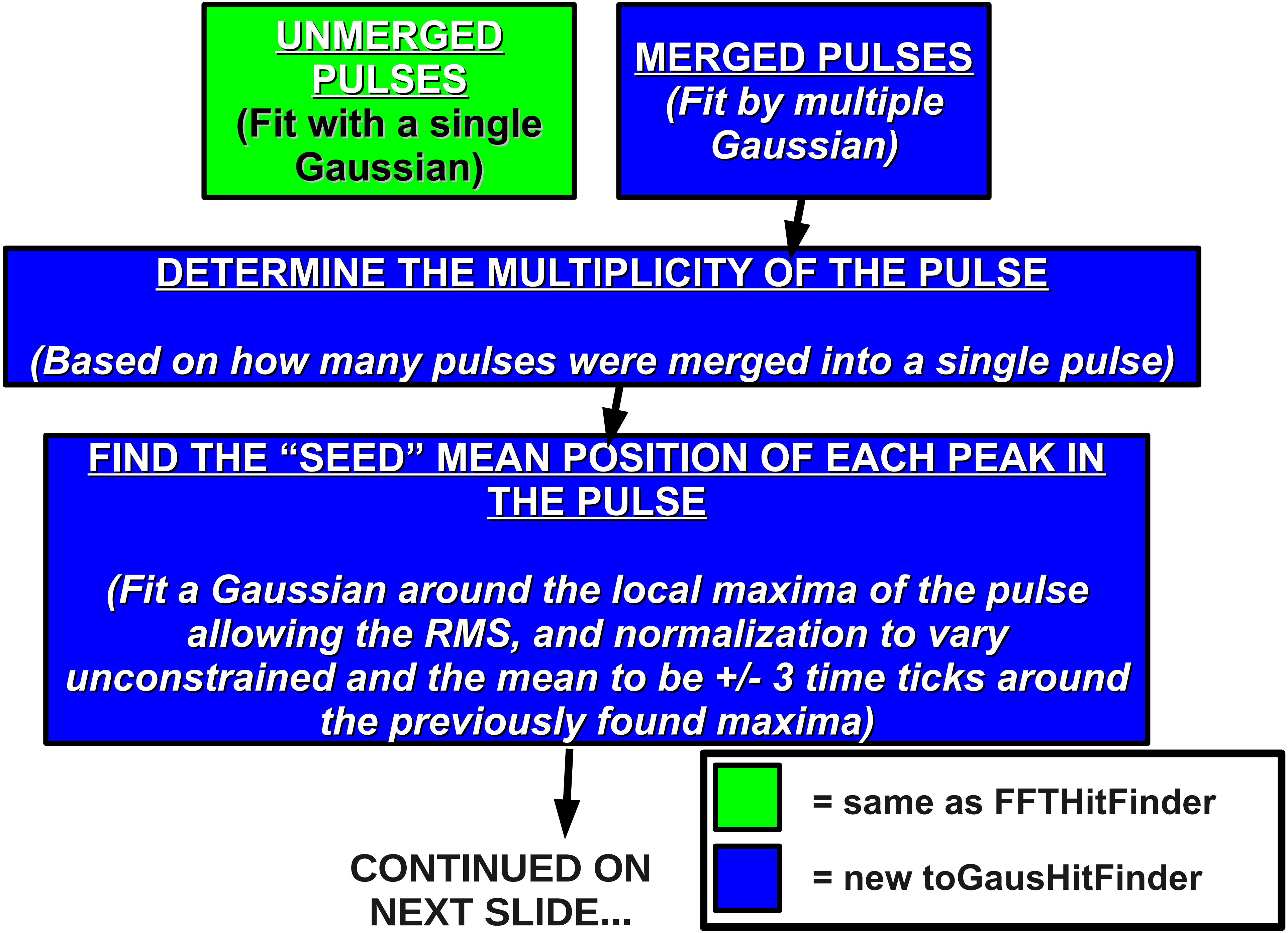
(We require the fit normalization to be > ½ the threshold and the fit RMS > minimum width)

WRITE OUT RECO:HIT

🟩 = same as FFTHitFinder

🟦 = new to GausHitFinder

GausHitFinder Algorithm



UNMERGED PULSES
(Fit with a single Gaussian)

MERGED PULSES
(Fit by multiple Gaussian)

DETERMINE THE MULTIPLICITY OF THE PULSE

(Based on how many pulses were merged into a single pulse)

FIND THE "SEED" MEAN POSITION OF EACH PEAK IN THE PULSE

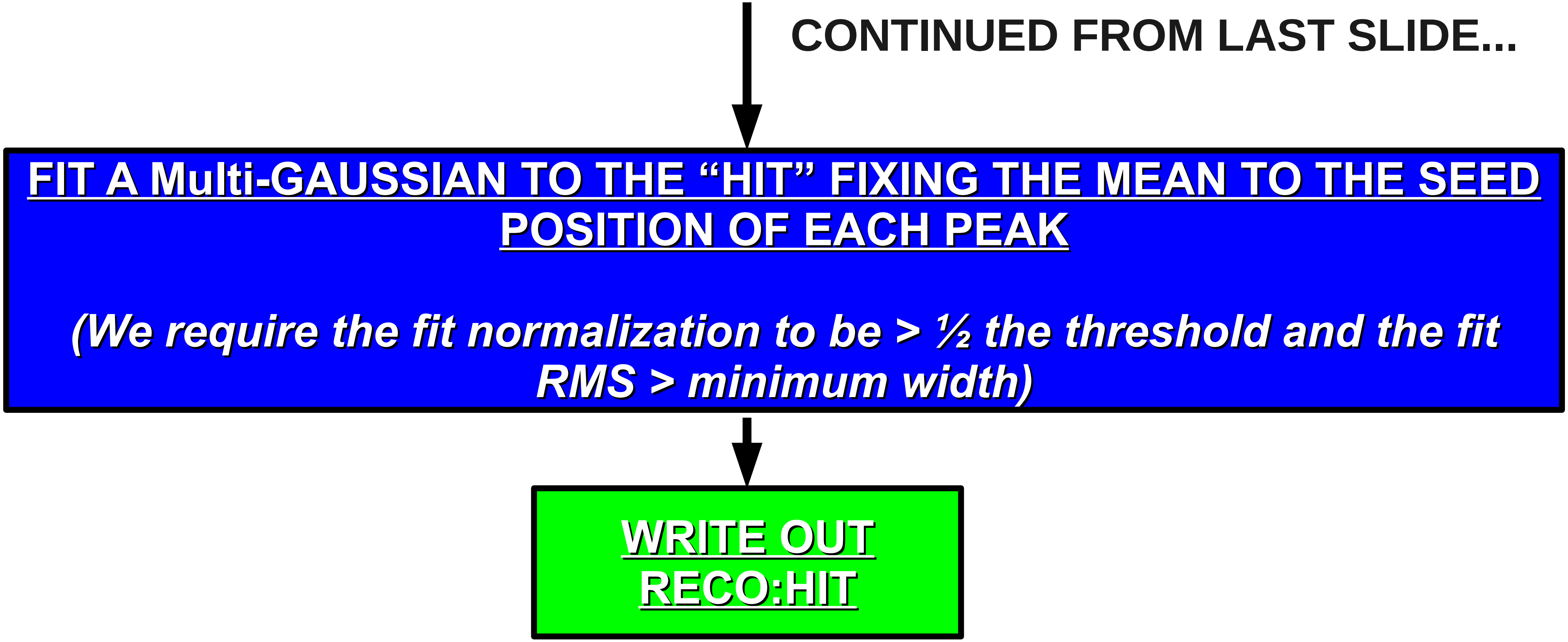
(Fit a Gaussian around the local maxima of the pulse allowing the RMS, and normalization to vary unconstrained and the mean to be +/- 3 time ticks around the previously found maxima)

CONTINUED ON NEXT SLIDE...

🟩 = same as FFTHitFinder

🟦 = new toGausHitFinder

GausHitFinder Algorithm



CONTINUED FROM LAST SLIDE...

FIT A Multi-GAUSSIAN TO THE “HIT” FIXING THE MEAN TO THE SEED POSITION OF EACH PEAK

(We require the fit normalization to be > ½ the threshold and the fit RMS > minimum width)

WRITE OUT RECO:HIT

Note: No rejection of hits based on χ^2/NDF is implemented yet

However, I would like to put this up for discussion